**Hadoop全自动化安装脚本**

# 1 设计目的

通过极其简单的方式，让开发者可以批量快速的构建大数据框架。

# 2 设计思路

## 2.1 ssh免密连接设计思路

为了简化ssh免密登录的过程步骤，官方给出了ssh-copy-id这种简单的命令工具，但是这种方法存在一种弊端，如果现在有一千台计算机，那么单纯的依靠此命令实现整个集群的免密登录，需要执行999\*999次该命令，这将严重影响系统的性能。故采用如下方法：

1. 先将所有的公钥通过ssh-copy-id命令发送给master主机，master自己也要发给自己一份。而这些公钥将会存储在.ssh/authorized\_keys这个文件中，然后在利用scp命令，将该文件发送给其他所有机器，这样计算机之间的连接次数将会减少至1999次，大大提高了传输效率。
2. Ssh命令中会出现手动输入密码的情况，交给except工具，实现自动输入
3. 利用while循环和wc -l命令，实时监控master这台机器上的authorized\_keys中的字符串个数，如果有1000条，说明接收了全部公钥，此时就可以利用scp命令执行分发操作了。

## 2.2 Hadoop自动化安装设计思路

完全按照Hadoop安装操作设计的脚本，只是简单地将此过程利用sed工具进行了修改。

## 2.3 conf配置文件设计思路

用户只需要在配置文件中简单的写入框架的主机IP，从机IP，以及登录密码，便可以直接搭建大数据框架。由于文件中包含敏感信息，权限建议设置为700。

# 3 具体的安装步骤

**所用的系统是centos7.0及以上版本，所用的包必须是hadoop-2.8.0.tar.gz，jdk-7u79-linux-x64.tar.gz。而且要把这些文件全部放入/root/目录下，因为我已经把这个写死了，如果换成其他安装包，或者放到其他路径，注意自己去修改脚本。**

1 修改conf文件，写入自己的主机IP和从机IP以及密码。

2 执行命令 bash ssh\_link.sh

然后就可以喝着茶，看系统自动给你搭建大数据框架了。整个过程不超过十分钟，简单省事。

# 4 注意事项

1 可能会出现环境变量没有生效的情况，保险起见，安装完成后，手动执行source /etc/profiles

2 如果slaver有多台，注意在conf文件中用引号括起来，并且每个IP用空格隔离开

3 请严格按照操作说明来，一定能完成安装，为了防止误操，建议设置快照，以便系统恢复

# 5 脚本缺陷

因为本人偷懒，并没有把安装包，安装路径写入到配置文件，故目前的脚本应用的还是固定的安装包，以及固定的路径，此处务必注意。

# 6 具体代码（已通过测试，放心用）

## 6.1 conf代码（改成你自己的IP和密码）

master=192.168.132.132

slaver=192.168.132.133

password=123456

## 6.2 ssh\_link.sh 代码

#!/bin/bash

#link every computer

#张孟辉 2046459994@qq.com

master=`cat -n /root/conf|grep master|sed s'/^.\*master=//g'`

slavers=`cat -n /root/conf|grep slaver|sed s'/^.\*slaver=//g'`

password=`cat -n /root/conf|grep password|sed s'/^.\*password=//g'`

localhost=`ifconfig | sed -n '2p'|sed 's/^.\*inet //g'|sed 's/ netmask.\*//g'`

yum -y install net-tools expect

slaver\_number=0

source /root/hadoop\_install.sh

for slave\_tempt in $slavers

do

slaver\_number=$(($slaver\_number+1))

done

expect -c "set timeout -1;

spawn ssh-keygen;

expect {

{\* in which to save the key \*} {send -- \r;exp\_continue}

{\*(empty for no passphrase)\*} {send -- \r;exp\_continue}

{\*same passphrase again\*} {send -- \r;exp\_continue}

eof {exit 0;}

}";

expect -c "set timeout -1;

spawn ssh-copy-id root@$master;

expect {

{\*(yes/no)\*} {send -- yes\r;exp\_continue}

{\*password\*} {send -- $password\r;exp\_continue}

eof {exit 0;}

}";

function testfile(){

#if [$master=$localhost];

#then

[ -f /root/.ssh/authorized\_keys ] && row\_number=`cat /root/.ssh/authorized\_keys | wc -l`||row\_number=0

}

function ssh\_all(){

for slave in $slavers

do

expect -c "set timeout -1;

spawn scp /root/.ssh/authorized\_keys root@$slave:/root/.ssh/;

expect {

{\*(yes/no)\*} {send -- yes\r;exp\_continue}

{\*password\*} {send -- $password\r;exp\_continue}

eof {exit 0;}

}";

done

}

if [ $master == $localhost ]

then

testfile

while [ $(($slaver\_number+1)) != $row\_number ]

do

echo '正在检查authorized\_keys这个文件是否接受收了全部公钥，十分钟之后还是这句话，请contr+c,终止程序，检查节点是否正常'

testfile

sleep 5

done

echo '接收完成，将要执行分发任务，请耐心等待'

ssh\_all

fi

echo '恭喜你，安装完成'

## 6.3 hadoop\_install.sh代码

#!/bin/bash

# it's a script to install hadoop

#张孟辉 2046459994@qq.com

#最近一次改动，修改slave写入异常问题

tar zxvf jdk-7u79-linux-x64.tar.gz

sed -i '9a export JAVA\_HOME=/root/jdk1.7.0\_79' /etc/profile

sed -i '10a export CLASSPATH=.:$JAVA\_HOME/jre/lib/rt.jar:$JAVA\_HOME/lib/dt.jar:$JAVA\_HOME/lib/tools.jar' /etc/profile

sed -i '11a export PATH=$PATH:$JAVA\_HOME/bin' /etc/profile

sed -i '12a export PATH=$PATH:/root/hadoop-2.8.0/bin:/root/hadoop-2.8.0/sbin' /etc/profile

source /etc/profile

tar zxvf hadoop-2.8.0.tar.gz

mkdir tmp dfs

mkdir dfs/data dfs/name

cd ./hadoop-2.8.0/etc/hadoop/

mv mapred-site.xml.template mapred-site.xml

sed -i '/^export JAVA\_HOME=${JAVA\_HOME}/ s/JAVA\_HOME=${JAVA\_HOME}/JAVA\_HOME=\/root\/jdk1.7.0\_79/g' hadoop-env.sh

sed -i '/^#.\*export JAVA\_HOME=.\*/ s/^.\*$/export JAVA\_HOME=\/root\/jdk1.7.0\_79/g' yarn-env.sh

sed -i '/^#.\*export JAVA\_HOME=.\*/ s/^.\*$/export JAVA\_HOME=\/root\/jdk1.7.0\_79/g' mapred-env.sh

sed -i '/<.\*configuration>/d' core-site.xml

cat >> core-site.xml << EOF

<configuration>

<property>

<name>fs.defaultFS</name>

<value>hdfs://$master:9000</value>

</property>

<property>

<name>hadoop.tmp.dir</name>

<value>file:/root/tmp</value>

</property>

<property>

<name>io.file.buffer.size</name>

<value>131702</value>

</property>

</configuration>

EOF

sed -i '/<.\*configuration>/d' hdfs-site.xml

cat >> hdfs-site.xml << EOF

<configuration>

<property>

<name>dfs.namenode.name.dir</name>

<value>file:/root/dfs/name</value>

</property>

<property>

<name>dfs.datanode.data.dir</name>

<value>file:/root/dfs/data</value>

</property>

<property>

<name>dfs.replication</name>

<value>2</value>

</property>

<property>

<name>dfs.namenode.secondary.http-address</name>

<value>$localhost:9001</value>

</property>

<property>

<name>dfs.webhdfs.enabled</name>

<value>true</value>

</property>

</configuration>

EOF

sed -i '/<.\*configuration>/d' mapred-site.xml

cat >> mapred-site.xml << EOF

<configuration>

<property>

<name>mapreduce.framework.name</name>

<value>yarn</value>

</property>

<property>

<name>mapreduce.jobhistory.address</name>

<value>$localhost:10020</value>

</property>

<property>

<name>mapreduce.jobhistory.webapp.address</name>

<value>$localhost:19888</value>

</property>

</configuration>

EOF

sed -i '/<.\*configuration>/d' yarn-site.xml

cat >> yarn-site.xml << EOF

<configuration>

<property>

<name>yarn.nodemanager.aux-services</name>

<value>mapreduce\_shuffle</value>

</property>

<property>

<name>yarn.nodemanager.auxservices.mapreduce.shuffle.class</name>

<value>org.apache.hadoop.mapred.ShuffleHandler</value>

</property>

<property>

<name>yarn.resourcemanager.address</name>

<value>$master:8032</value>

</property>

<property>

<name>yarn.resourcemanager.scheduler.address</name>

<value>$master:8030</value>

</property>

<property>

<name>yarn.resourcemanager.resource-tracker.address</name>

<value>$master:8031</value>

</property>

<property>

<name>yarn.resourcemanager.admin.address</name>

<value>$master:8033</value>

</property>

<property>

<name>yarn.resourcemanager.webapp.address</name>

<value>$master:8088</value>

</property>

<property>

<name>yarn.nodemanager.resource.memory-mb</name>

<value>4096</value>

</property>

</configuration>

EOF

if [ $master == $localhost ]

then

for slave\_i in $slavers

do

sed -i '$ a\'$slave\_i'' slaves

done

fi

sed -i '/localhost/d' slaves

#echo 'zhang2' >> slaves

# slave important!!!!